FAULT-TOLERANT COMPUTATION IN SEMIGROUPS AND SEMIRINGS

Master of Engineering Thesis

Christoforos N. Hadjicostis

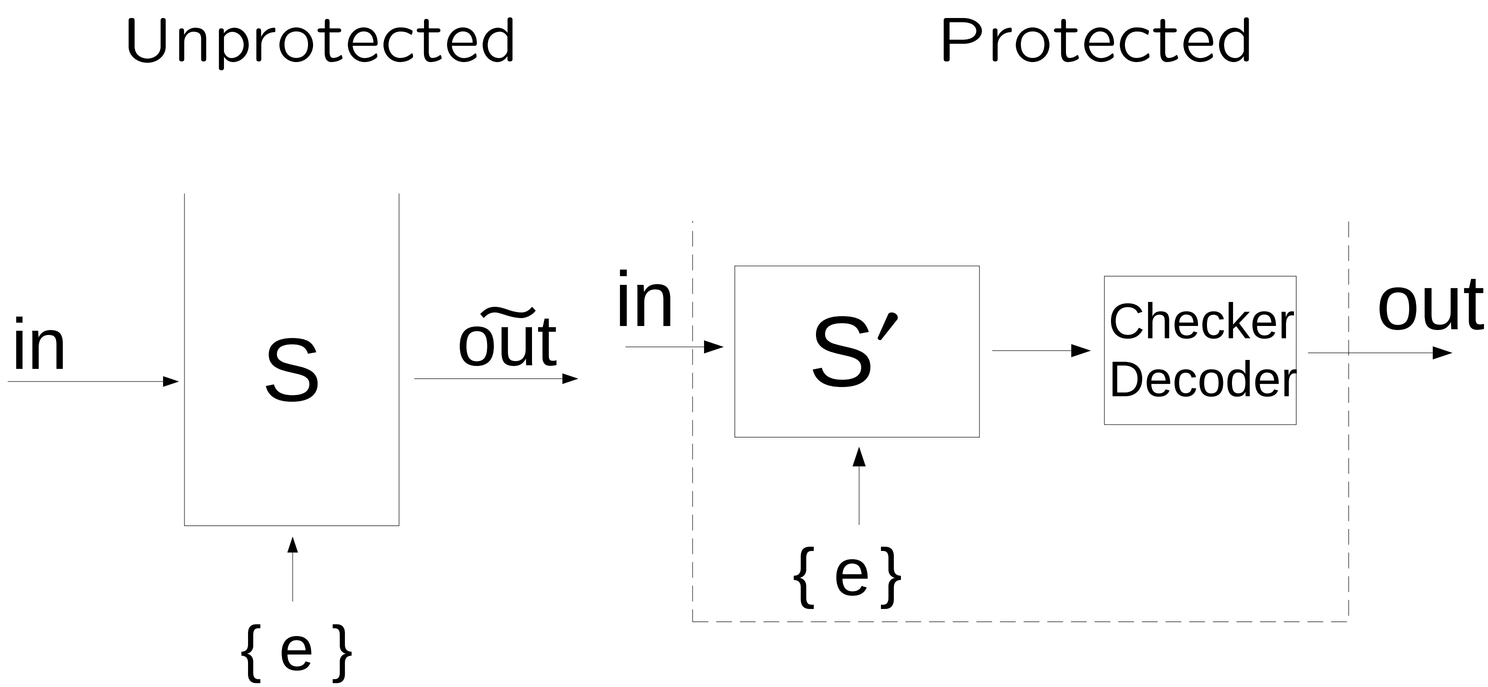
Supervisor: Prof. George C. Verghese

May 1995

Plan

- Introduction and Definitions
- Fault Tolerance for Semigroup-Based Computations
- Extensions and Future Possibilities
- Contribution

Fault-Tolerant Computational Systems



$$\widetilde{\text{out}} = f\,(\text{in}, \{e\})$$

Arithmetic Codes $\longleftrightarrow$ ABFT

(... Diamond, 1955 $\longleftrightarrow$ Abraham, 1984 ...)

$n \times n$ Matrix Multiplication

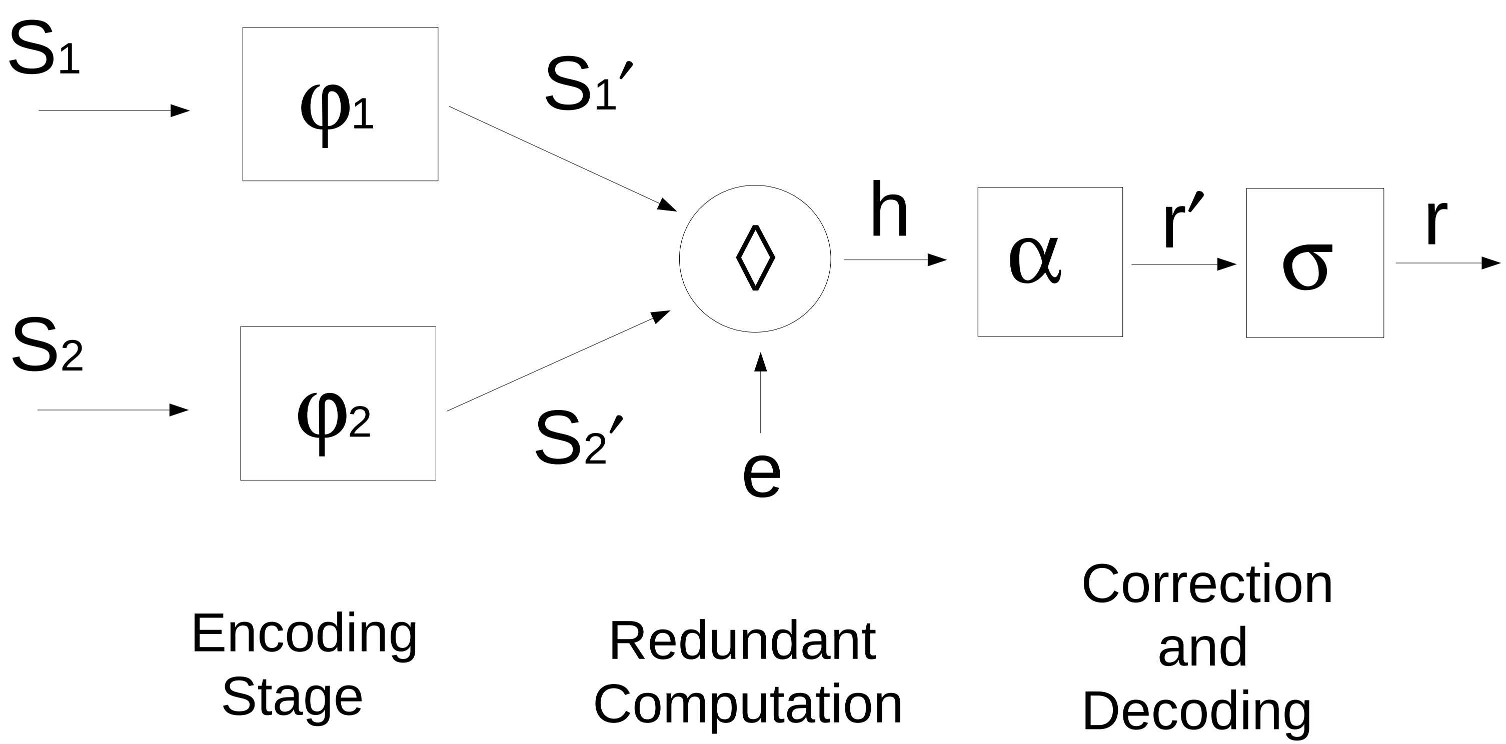
(Abraham et al., 1984 onwards)

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{bmatrix}$$

$$\begin{bmatrix} & A & & \\ c_1 & c_2 & \dots & c_n \end{bmatrix} \times \begin{bmatrix} & r_1 \\ & r_2 \\ B & \vdots \\ & r_n \end{bmatrix} = \begin{bmatrix} & & & rs_1 \\ & & & rs_2 \\ & C & & \vdots \\ & & & rs_n \\ cs_1 & cs_2 & \dots \; cs_n & ? \end{bmatrix}$$

Model for Fault-Tolerant Group/Semigroup Computation

(Group Case by Beckmann, MIT, PhD, 1992)



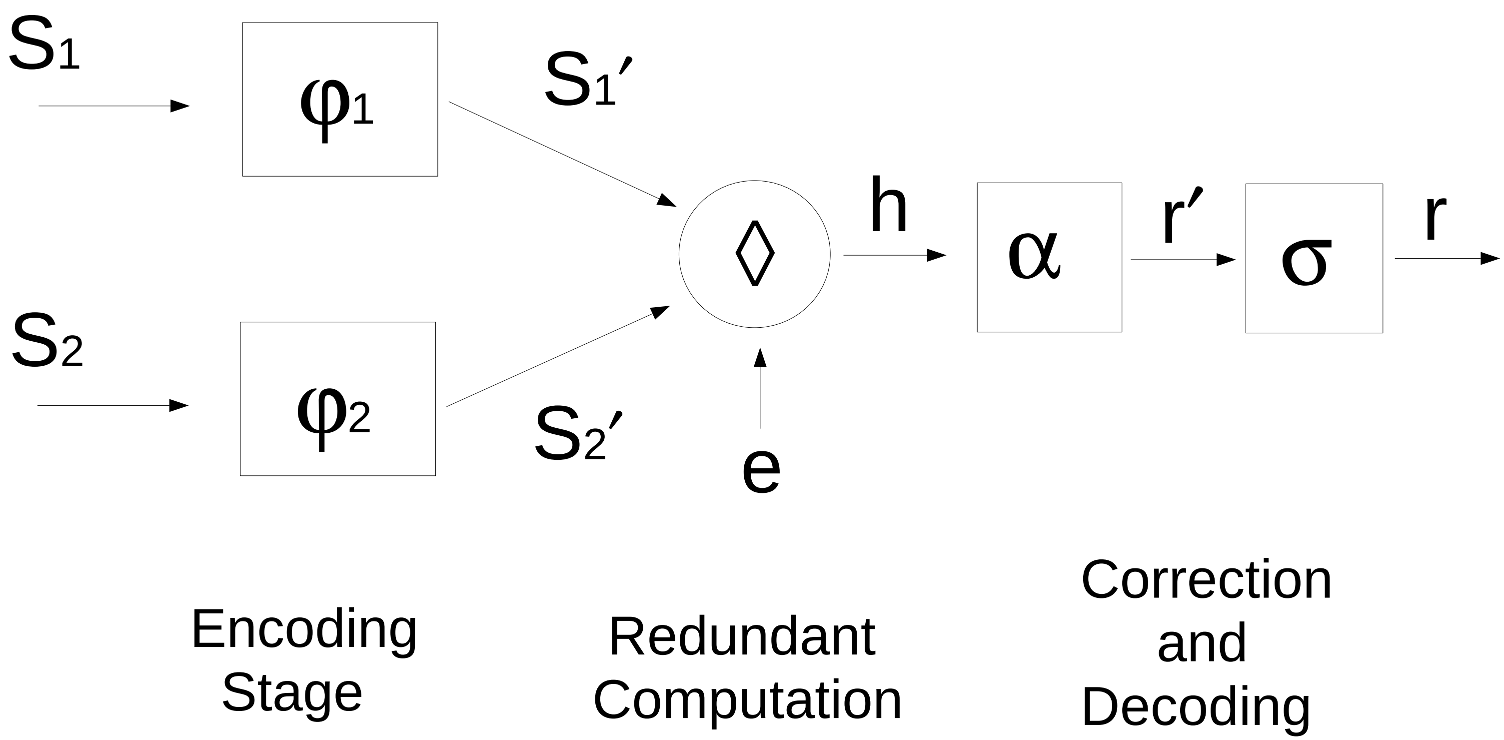
Definition of a Semigroup

Definition: A semigroup $\mathcal{S} = (S, \circ)$ is a set of elements S, and a binary operation $\circ$, called the *semigroup product*, such that:

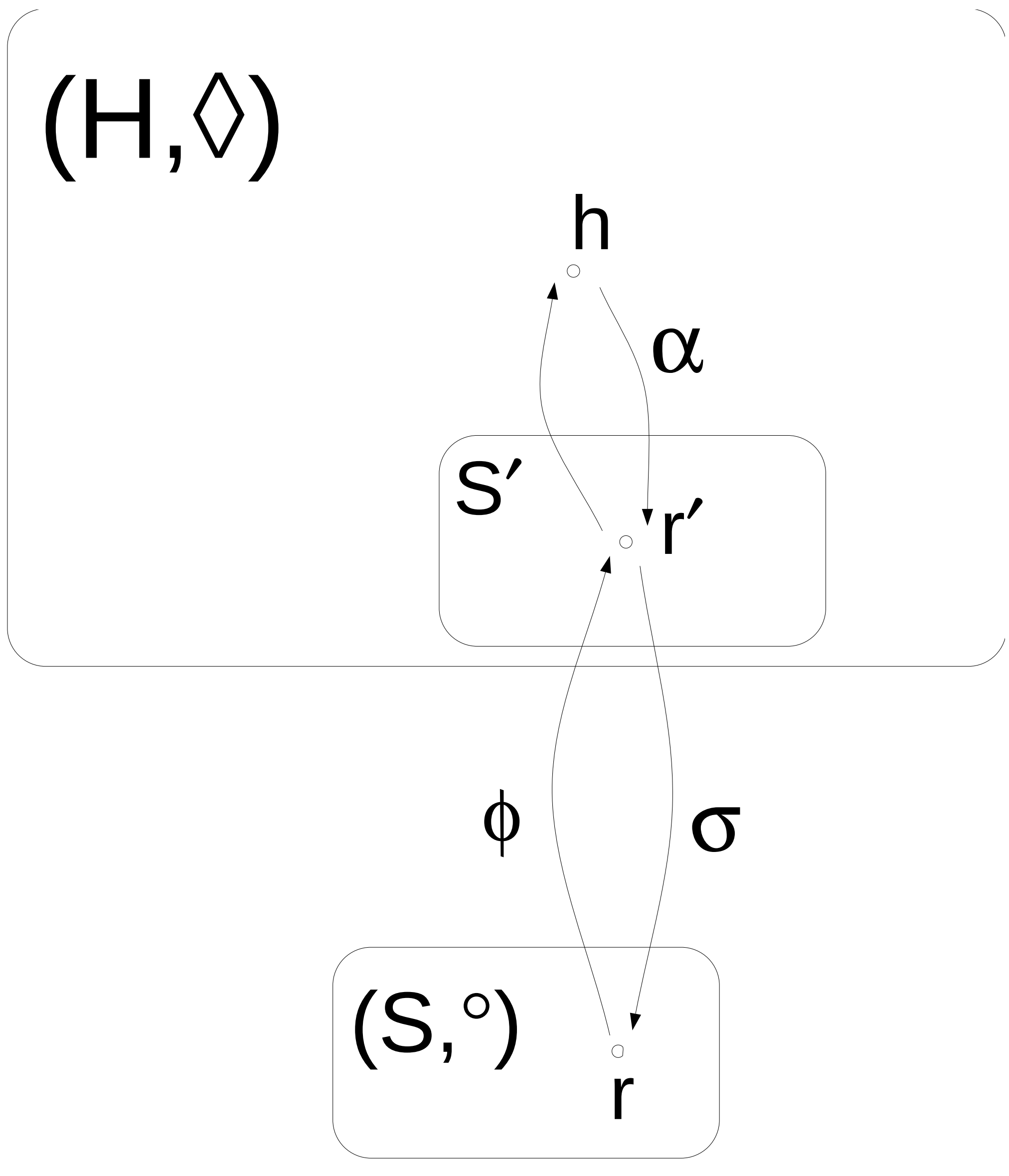
1. For all $s_1, s_2 \in S$, $s_1 \circ s_2 \in S$
2. For all $s_1, s_2, s_3 \in S$, $(s_1 \circ s_2) \circ s_3 = s_1 \circ (s_2 \circ s_3)$

Examples: $(Z, \times)$, $(N, \times)$, $(N_0, +)$

Abelian Semigroup Computation



Use of Abelian Semigroup Homomorphisms



Separate Codes

Use a separate channel (known as “parity” channel):

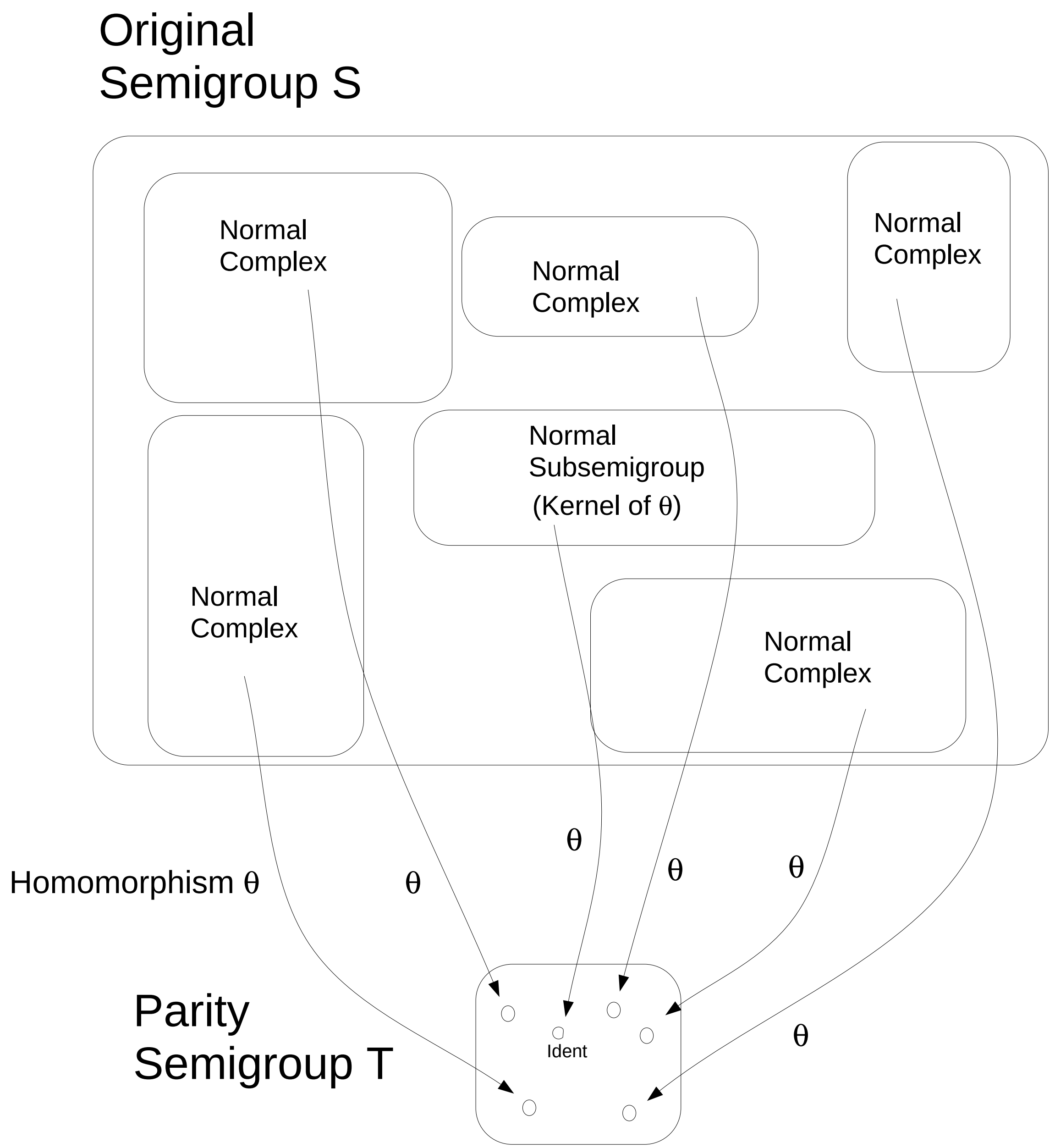
$$
\begin{aligned}
s_1 &\stackrel{\phi}{\longmapsto} (s_1, t_1) \\
s_2 &\stackrel{\phi}{\longmapsto} (s_2, t_2) \\
s_1 \circ s_2 &\stackrel{\phi}{\longmapsto} (s_1 \circ s_2, t_1 \odot t_2) = (s_1, t_1) \diamond (s_2, t_2)
\end{aligned}
$$

Parity encoding mapping θ

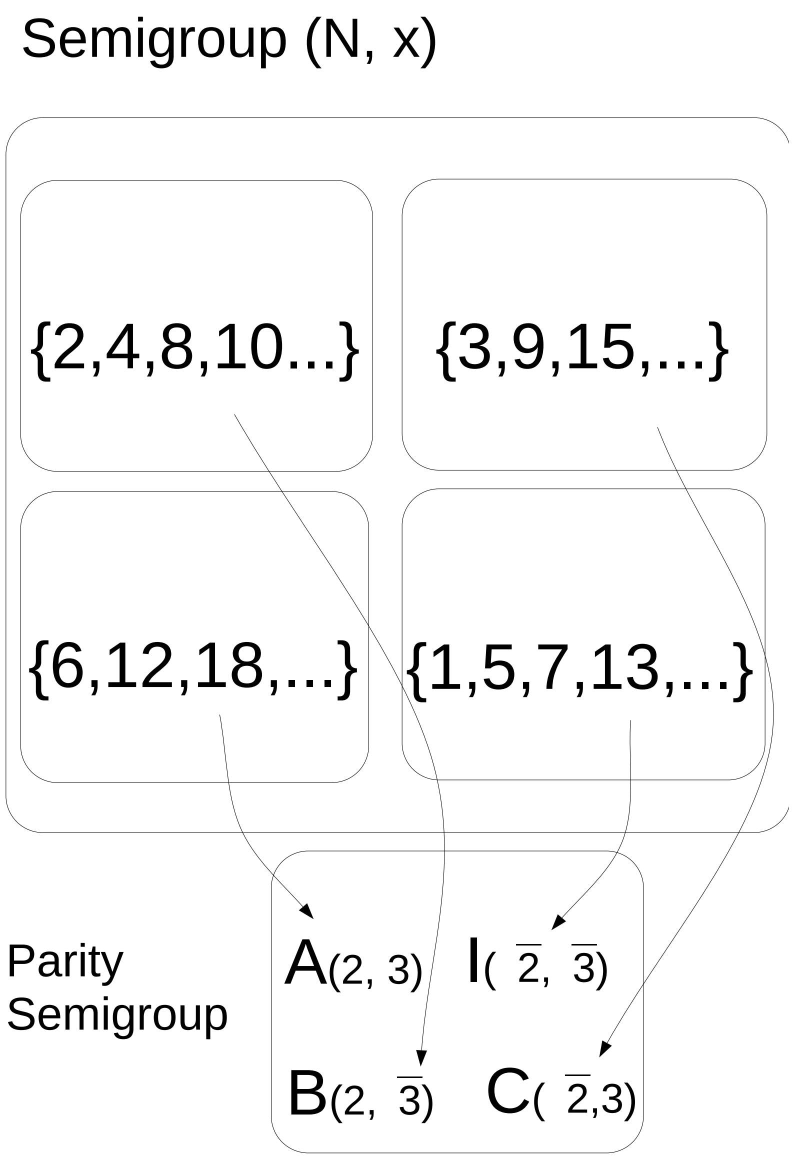
$$
\begin{aligned}
t_1 &= \theta(s_1) \\
t_2 &= \theta(s_2)
\end{aligned}
$$

θ is a semigroup homomorphism.

Separate Codes in Semigroups



Separate Codes for $(N, \times)$



$\odot$	I	A	B	C
I	I	A	B	C
A	A	A	A	A
B	B	A	B	A
C	C	A	A	C

Extensions and Future Work

- More complicated algebraic structures can be studied. (Already done for semiring computations, such as $(N_0, +, \times)$, and $(Z, max, +)$)

- Future Research
  1) Reflect hardware failure modes
  2) Study of error correction procedures (including distance metrics, syndromes, etc.)
  3) Protect strings of computations.
  4) Links to error-correcting coding and automata theory.

Contribution

- Algebraic framework for fault-tolerant semigroup computations

- Separate codes are completely characterized

- Explored various examples